SANDIA REPORT

SAND2002-3237
Unlimited Release
Printed October 2002

Self-Reconfigurable Robots

David M. Hensinger, Gabriel A. Johnston, Elaine M. Hinman-Sweeney, John Feddema, and Steven Eskridge

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2002-3237
Unlimited Release
Printed October 2002

SELF-RECONFIGURABLE ROBOTS

David M. Hensinger, Gabriel A. Johnston, and Elaine M. Hinman-Sweeney
Advanced Engineering and Manufacturing Software Development

John Feddema and Steven Eskridge
Intelligent Systems Controls

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1010

Abstract

A distributed reconfigurable micro-robotic system is a collection of unlimited numbers of distributed small, homogeneous robots designed to autonomously organize and reorganize in order to achieve mission-specified geometric shapes and functions. This project investigated the design, control, and planning issues for self-configuring and self-organizing robots. In the 2D space a system consisting of two robots was prototyped and successfully displayed automatic docking/undocking to operate dependently or independently. Additional modules were constructed to display the usefulness of a self-configuring system in various situations. In 3D a self-reconfiguring robot system of 4 identical modules was built. Each module connects to its neighbors using rotating actuators. An individual component can move in three dimensions on its neighbors. We have also built a self-reconfiguring robot system consisting of 9-module Crystalline Robot. Each module in this robot is actuated by expansion/contraction. The system is fully distributed, has local communication (to neighbors) capabilities and it has global sensing capabilities.

Contents

List of Figures

I. Introduction

Distributed reconfigurable micro-robotic systems are still academic curiosities. Three types of modular reconfigurable robotic systems have been proposed in the literature: (1) Robots in which modules are reconfigured using external intervention (e.g., [BeZL89, CoLDB92, Sci85, Wu86]), (2) Cellular robotic systems in which a heterogeneous collection of independent specialized modules are coordinated (e.g., [Be88, BeW91, FuN88, FuK90]), and (3) Swarm intelligence in which there are generally no physical connections between modules. More recently, two other types of modular reconfigurable robotic systems have been considered. [Yim93, Yim94] considered modular robots composed of a few basic elements, which can be composed into complex systems, and used for various modes of locomotion. [MuKK94, MuKK95] considered a 'fractal' system composed of modules with zero kinematic mobility, but which can 'walk' over each other in discrete quanta due to changes in the polarity of magnetic fields. The first hardware realized systems date back to 1994 by Murata in Japan. He developed a planar self-assembling system in 2-D space that used electromagnets to connect three homogeneous mechanical units. An infrared optical device was used to communicate between modules when connected. There was no sensing between mechanical units when not connected, and therefore units would connect and reconnect at random until they received correct communication signals. Currently Rus at Dartmouth College is building a system of "Molecules", which are also linked together by permanent magnets. They have simulated several of these modules interconnecting to form larger truss-like structures but have yet to verify their design with hardware.

This project investigated two classes of reconfigurable homogeneous micro-robotic systems. The first system considered reconfigurable robotic modules in 2-D space while the second system extended this approach with the design and development of reconfigurable units in 3-D space. Upon command, the individual modules can autonomously search for other elements and connect together both electrically and mechanically to form composite structures (composite in both an electrical and mechanical sense) suitable for accomplishing specific functions and tasks. When combined with several modules, the constructed machine will be able to perform tasks, which individual modules could not perform alone. Mechanically, the individual modules could group together to form large wheels or other locomotion elements applicable to traversing rough terrain or obstacles.

II. 2D Adapted Gemini Design

The ISRC's robotic platform, Gemini was adopted for the 2-D space design. The original Gemini platform is a highly mobile vehicle with a passive joint between modules. It consists of two tracked robots, which were passively connected at a center joint. It was demonstrated that a dramatic increase in mobility was achieved through having two separate vehicles with a joint in the middle rather than having one small robot or one large robot. The small robot could not get over large obstacles while the large robot lacked in flexibility. Gemini turned out to be the best of both worlds.

The passive joint was redesigned by incorporating a quick-disconnect mechanism utilizing rare-earth magnets to attach modules, an articulating joint allowing three types of joint conditions, and a release mechanism allowing Gemini bodies to intentionally disconnect and provide automatic separation in the event of power failure of a particular body. The new versions of the Gemini robots were named Castor and Pollux shown in Figure 1.

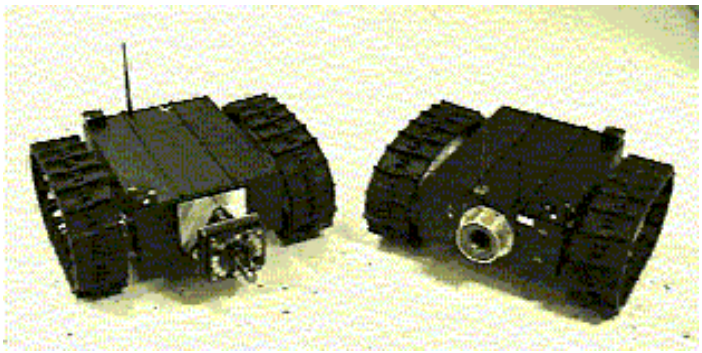


Figure 1. Castor and Pollux

Linkage

The 2D reconfigurable vehicle extends the Gemini design by providing a latching/release mechanism and two controlled degrees-of-freedom between the two vehicles. The connect/disconnect feature was added so that both robots could work independently but when posed with a problem that was better suited for a larger robot they could dock. An example of this might be two robots out in a field performing a function. They could work independently until they came to an obstacle too large for them to cross. They could then dock and effectively double their size to cross the ditch without falling in. On the other side of the obstacle, they disconnect and continue to work independently.

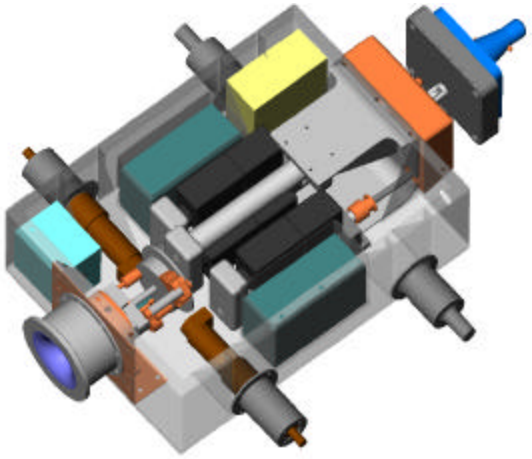


Figure 2. CAD Model of 2D Reconfigurable Robot

Preliminary studies of the center connect/disconnect joint had been done by manually driving one robot into the other and manually running code that had been written to connect the two. After this proved to be a promising feature it was suggested that sensors be added to allow the vehicles to autonomously dock. In addition, each vehicle has both

a male and female end for connecting to two other vehicles, so that multiple vehicles can connect to each other and create a single train-like vehicle with even greater mobility.

On the male end of the vehicle are several permanent magnets that provide 230 pounds of force between two vehicles when they are engaged. On the female end of the vehicle is a release mechanism that pushes the male side of the other vehicle far enough away that the permanent magnets lose contact with the female side of the vehicle, which results in substantially less magnetic force. This release mechanism is a fail-safe disconnect, signifying that if power is lost on one vehicle, the release mechanism will push the other vehicle away so that they are no longer connected. In this way, the active vehicle can continue to perform its mission without having to drag around the inactive vehicle. There are two stages involved in connection. During stage 1, male and female quick-disconnect couple, allowing the ring of rare earth magnets on the male end to attach to a metal flange on the female end. During stage 2, the female flange rotates passively on bearing allowing spring-loaded alignment pins on the male end to snap into place. This action freezes the rotational degree of freedom. Decoupling the components and allowing them to function autonomously can obviate any degradation of the performance of the coupled system.

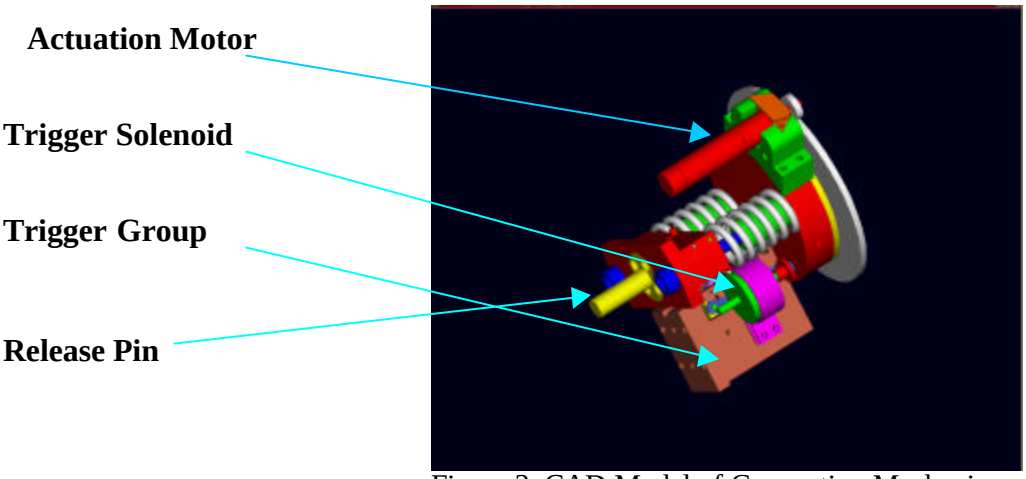


Figure 3. CAD Model of Connection Mechanism

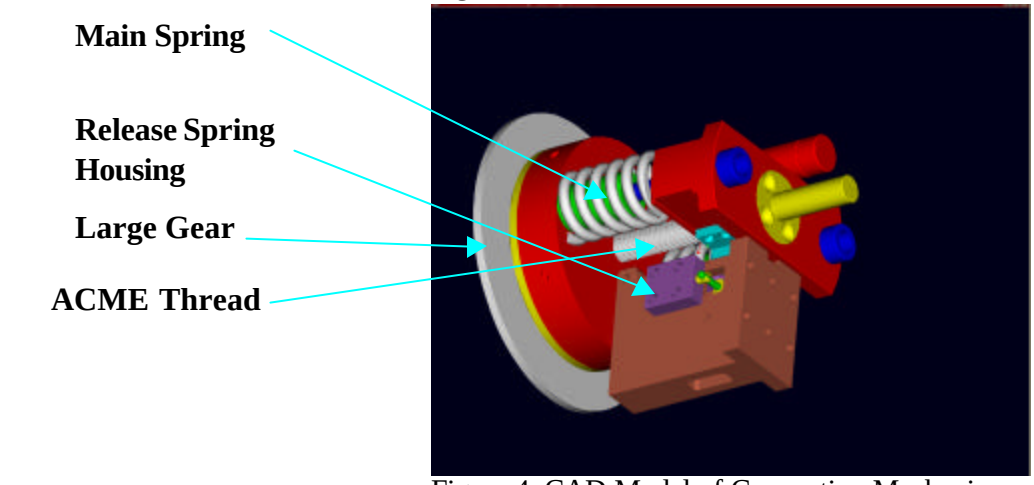


Figure 4. CAD Model of Connection Mechanism

The release mechanism works by storing approximately 250 pounds of force in the main springs (see Figures 3 and 4). This force is held by a sear, which is in turn held by a solenoid. Turning power off releases the spring force, and the male and female ends of the quick detach mechanism are forced to separate by the release pin.

In the initial testing phase of this system we determined that the actual loading on the magnetic quick-disconnect in certain situations was in excess of the design load. Furthermore, we learned that any sort of mechanism using magnets would not have sufficient holding force to overcome this dynamic loading. Our solution was to replace the magnetic system with a much stronger mechanical locking system.

Multiple Configurations

Two degrees-of-freedom on the male end of the reconfigurable vehicle are controlled with three linear actuators. Three push rods attached to the actuators push against a plate that is connected to the vehicle with a passive U-joint.

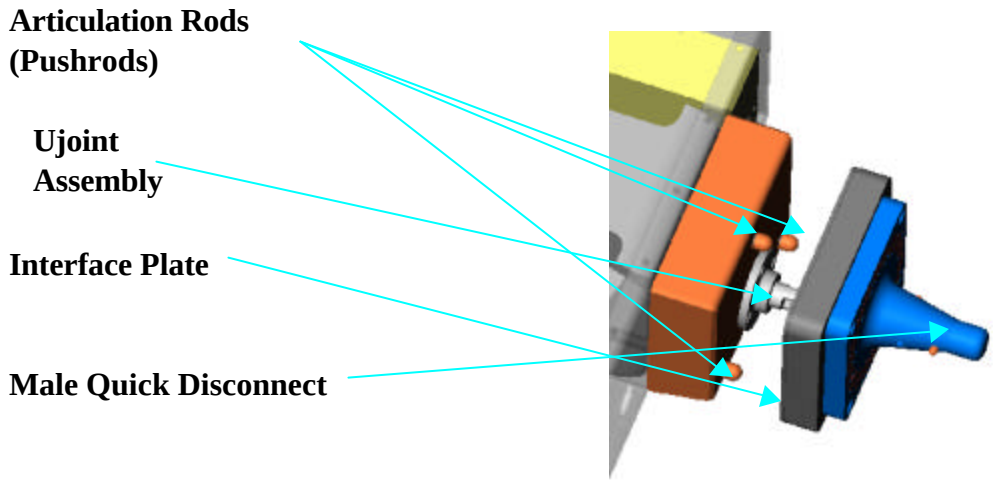


Figure 5. Close-up of Male End Showing Pushrods

If the three linear actuators are fully retracted, then the two degrees-of-freedom are free to rotate. This mode of operation maximizes traction on a rough terrain by allowing the treads of each vehicle to conform to the landscape. Driving the three linear actuators so that they contact the plate, the two degrees-of-freedom can be precisely controlled. In this mode, the vehicles can be articulated so that they form a vertical V-shape, which shortens the wheelbase and makes it easier to rotate about a position.

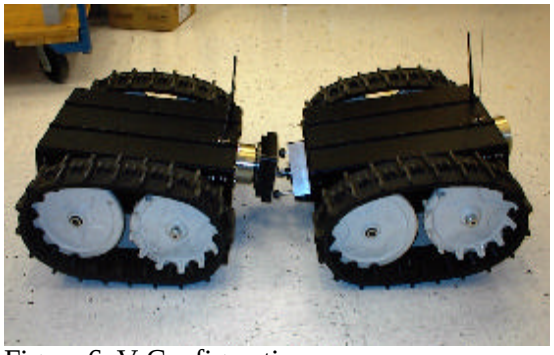


Figure 6. V Configuration

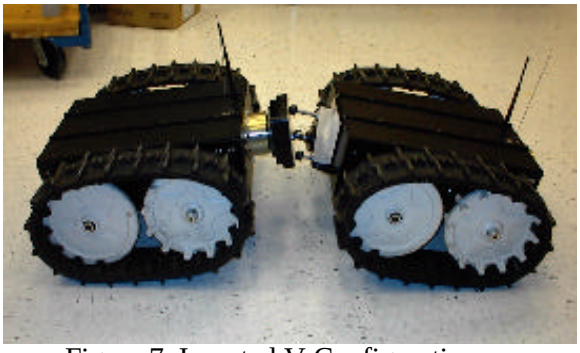


Figure 7. Inverted V Configuration

Similarly, the actuators can be articulated in an inverted V-shape to get over obstacles such as railroad ties. In the horizontal direction, the relative position of the vehicles can be placed in a "power-turn" configuration that allows the reconfigurable vehicle to turn with all treads moving forward. This is especially useful for turning while going up steep inclines. In performance trials the coupled Gemini platform is able to adopt 6 distinct static configurations. The static configurations have unique implications for the mobility and mission effectiveness of the coupled system. The effect these configurations have on range, mobility, maneuverability, observability, and stability of the coupled system is a subject of current evaluation. Configurations have been identified that improve the range and mobility of the coupled system over that of the uncoupled system.

III. Autonomous Docking

A docking sensor subsystem that automatically guides the connection process between two vehicles was developed. Infrared was one of the sensors considered for this project. IR sensors were not chosen for two reasons. One, these robots were designed to work both indoors and outdoors. If a piece or mud were to be splashed on the sensor the vehicles would not be able to dock. Two, the sun naturally produces infrared light and would change sensor readings between indoor and outdoor applications. Something more robust was needed.

If light did not turn up to be a solution the next natural choice was sound. This led to ultrasonic transducers. These sensors were larger than the IR sensors but were less affected by dirt, mud and dust. These sensors also gave the same output readings indoor and outdoor. For these reasons ultrasonic transducers were chosen for this project.

Previous research in docking using IR sensors accomplished this by using one transmitter and two receivers. They were comparing output from the receivers and calculating where the transmitter was located using a triangulation. This same method could have been used on this project by simply substituting ultrasonic transducers for the IR sensors. The IR sensor was needed since differential GPS will only accurately guide the vehicles to within one meter of each other while the connecting mechanism requires a precision of 100 centimeters. We have considered optical, laser, and ultrasound sensors for guidance, and we have concluded that ultrasound provides the simplest, most robust, low cost solution. Testing with narrow-beam ultrasound transducers by Airmar Technology Corporation looks promising, and we are currently developing the subsystem electronics

required to perform the docking. At the same time, we will be adding the GPS and compass subsystems used in the RATLER vehicles. This will require no new additional software since all electronic components in the reconfigurable vehicles are compatible with the RATLER vehicles.

States

A simulation was written to determine where the docking limits were and to help in the optimization of the docking procedure.

When running the simulation or the autodock.m file, the program prompts the user to "Enter current Pollux position and heading (DEG): [x y phi] ". After doing so, the program then loads all necessary information and using a fourth order Runge-Kutta ODE solver, calls the file pollux.m. Pollux.m has all the vehicle dynamics and calculates the position the vehicle should head. This process will continue for a pre-determined set of time. After the first vehicle, Pollux has moved to its final destination, the program saves its final position and again using the ODE solver calls the program castor.m. This program is very similar to pollux.m with exception that now Pollux is fixed and Castor moves.

The simulation uses a fourth-order Runge-Kutta [1] method for finding incremental steps in the x and y positions.

These slope estimates, k_1, k_2, k_3, and k_4, are calculated as follows:

$k_1 = f(t_{j-1}, y_{j-1})$, (1)
$k_2 = f(t_{j-1} + h/2, y_{j-1} + h/2*k1)$, (2)
$k_3 = f(t_{j-1} + h/2, y_{j-1} + h/2*k2)$, (3)
$k_4 = f(t_{j-1} + h, y_{j-1} + h*k3)$. (4)

A weighted average of the k_i are used to compute the next value of y:

$y_j = y_{j-1} + h(k_1/6 + k_2/3 + k_3/3 + k_4/6)$. (5)

Inside pollux.m and castor.m the heart of determining where the vehicle should move lies in the state that the vehicle is in. There are six states that the vehicle may be operating in. State 1 is the swing mode, state 2 is the sample mode, state 3 is the hswing mode, state 4 is the closer mode, state 5 is the find sensor mode, and state 6 is the closer2 mode. These states are named after actual variable names in the Galil motion controller program present on both robots. A graphical representation of states and the docking procedure follows the brief descriptions below:

State 1 or the swing mode turns the robot 30 degrees so that sampling may begin.

State 2 or the sampling mode checks to see if the distance between both robots is close enough and if the angle between them is small enough that the ultrasonic transducers would pick up a signal from the other. These parameters are modeled after actual

hardware. The sampling mode makes a 60-degree sweep from the starting position of 30 degrees to –30 degrees. Note that if the robots are within sensor parameters the mode will be switched to mode 5, find sensor mode.

State 3 or the hswing mode returns the robot back to the home position if the sensor was not found.

State 4 or the closer mode moves the robot forward by a couple of feet so that the steps one through three can repeat.

State 5 or the find sensor mode happens only when all the right parameters are met in mode 2. If they are, then the robot proceeds to rotate until the strongest signal found which also aligns the robot for docking.

State 6 or the closer2 mode moves the robot toward the other robot by a couple of feet only after it has aligned itself with the other robot.

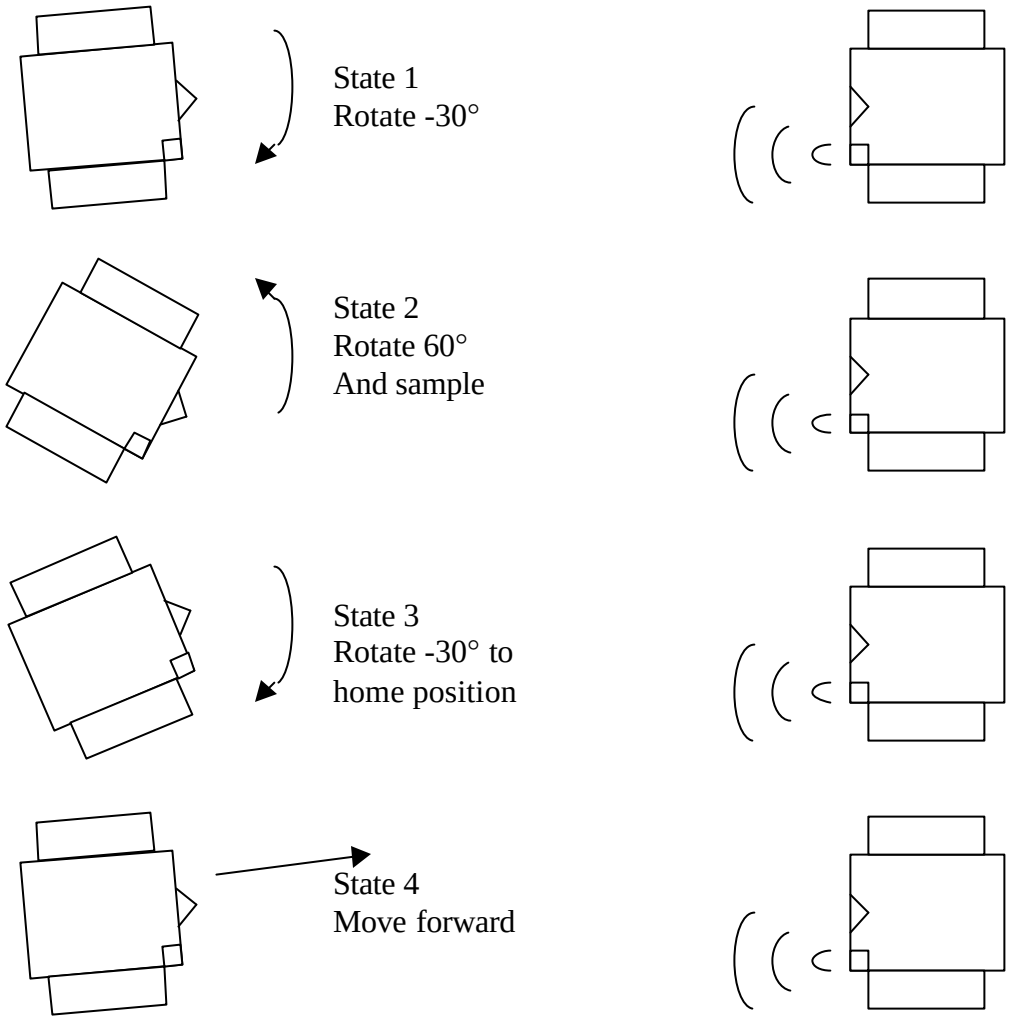


Figure 8. Graphical Representation of States

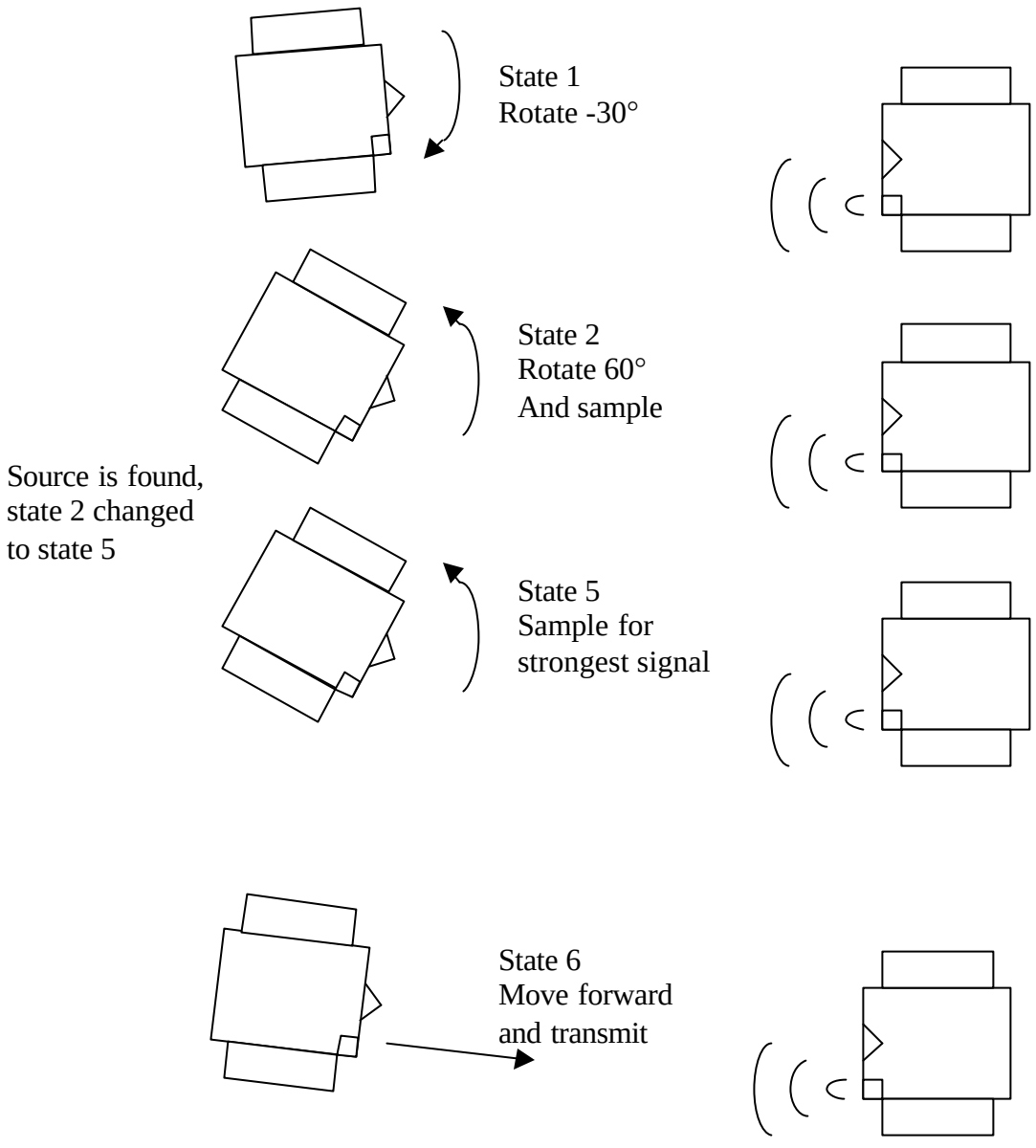


Figure 8 Continued. Graphical Representation of States

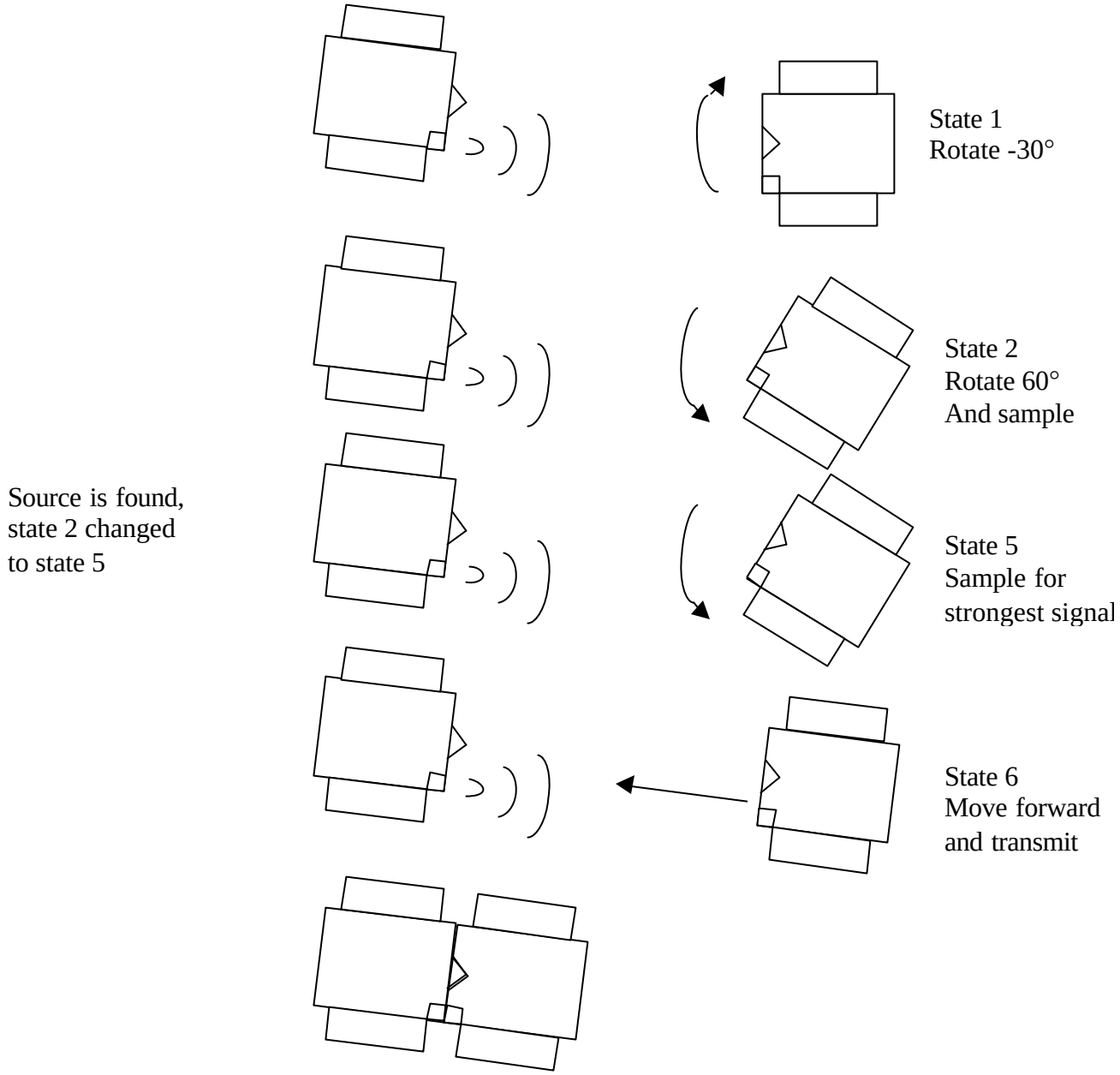


Figure 8 Continued. Graphical Representation of States

After the program determines which state it is in, it calculates the x, y, and heading error. It then multiplies this error times the Jacobian of the robot. The Jacobian for this robot is:

$$J = \begin{bmatrix} R*\cos q & R*\cos q \\ R*\sin q & R*\sin q \\ -2*R/W & 2*R/W \end{bmatrix}$$

Where R is the wheel radius, W is the wheelbase and θ is the vehicle heading.

The basic equation for determining where the robot should go is

$$\mathbf{Xdot} = J*\mathbf{U} \qquad (6)$$

Where **Xdot** and **U** are vectors and **U** is the right and left wheel velocities.

Because the Jacobian is not square, some matrix manipulation is required to find **Xdot**.

$$J^T*\mathbf{Xdot} = (J^T*J)*\mathbf{U} \qquad (7)$$

$$(J^T*J)^{-1}*J^T*\mathbf{Xdot} = \mathbf{U} \qquad (8)$$

Let,

$$J^+ = (J^T*J)^{-1}*J^T \qquad (9)$$

J^+ is sometimes referred to as the Moore-Penrose pseudoinverse or pinv.

The control portion used in the simulation to calculate U is

$$\mathbf{U} = J^+*\mathbf{X^d dot} \qquad (10)$$

Where,

$$\mathbf{X^d dot} = \Delta x = (x^d - x,\ y^d - y,\ \theta^d - \theta)^T$$

Now plugging (10) back into (6) to get

$$\mathbf{Xdot} = J*\ J^+*\mathbf{X^d dot} \qquad (11)$$

Simulation Results

There are two sets of results that will be examined. The first is an in depth look at a single case. The user inputs a current vehicle position and the simulation plots several graphs and information on heading, state, etc. The second set uses that same start position and creates a starting position array to find docking limits.

Single Case Results

Below is a brief description of each of the plots autodock.m generated. The plots will follow immediately after that. The initial position and heading used was [0 1 5].

Figure 9 plots a top view of the coordinate system Pollux is moving on. It can be seen that the vehicle moves straight until approximately (4,1.5) where it senses Castor and changes its heading.

Figure 10 plots the angle Pollux is facing. An initial heading was given as 5 degrees. The vehicle is preprogrammed to turn 30 degrees, then sample for 60 degrees in the opposite direction. The plot shows how the vehicle turns to –25 degrees then samples to 35 degrees. If Castor was not found then the vehicle is supposed to rotate back to its starting position and move forward and repeat the process.

Figure 11 shows Pollux's Y position and shows how the vehicle does not move while it is sampling but does after returning to the home position if Castor is not found.

Figure 12 shows Pollux's X position, shows how the vehicle does not move while it is sampling but does after returning to the home position if Castor is not found.

Figure 13 shows how the vehicle changes states:

State 1: Swing - The vehicle is turning to begin sampling.
State 2: Sample - The vehicle is sampling for Castor.
State 3: Swing - Castor was not found, vehicle is returning to its initial heading.
State 4: Closer - Vehicle is moving forward.
State 5: Find Sensor - Castor found, sample for strongest signal.
State 6: Closer2 - Vehicle aligned with Castor, move forward for docking.

Figures 14-19 are the same as Figures 9-13 with the exception that Castor is moving towards Pollux. It can be seen that Castor very quickly aligns with Pollux since Pollux is already in close proximity to Castor.

Figure 19, X-Y position of both robots, shows the results of Figure 9 and Figure 14 together. It can be seen that both robots successfully dock at approximately (6,0.5).

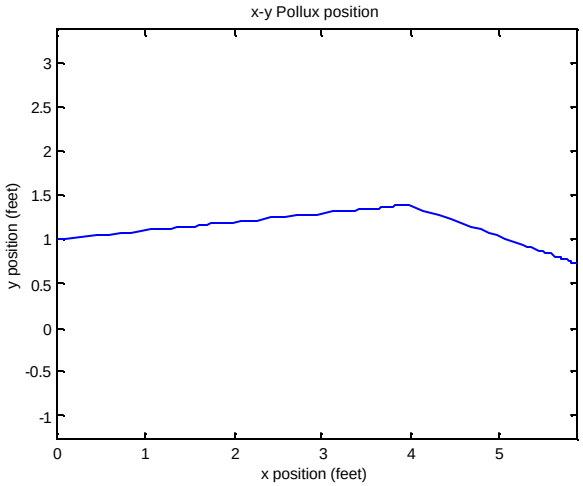


Figure 9. X-Y Pollux Position

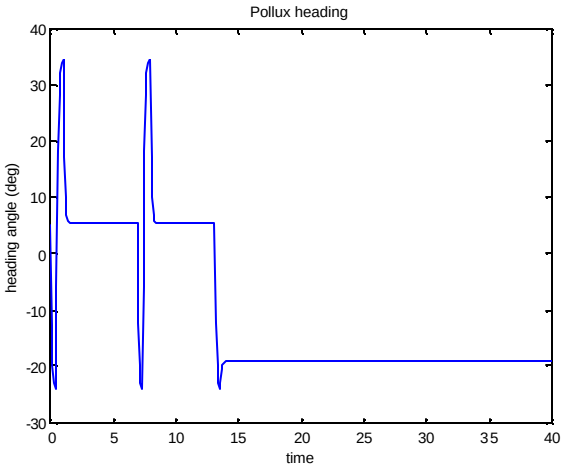


Figure 10. Pollux Heading

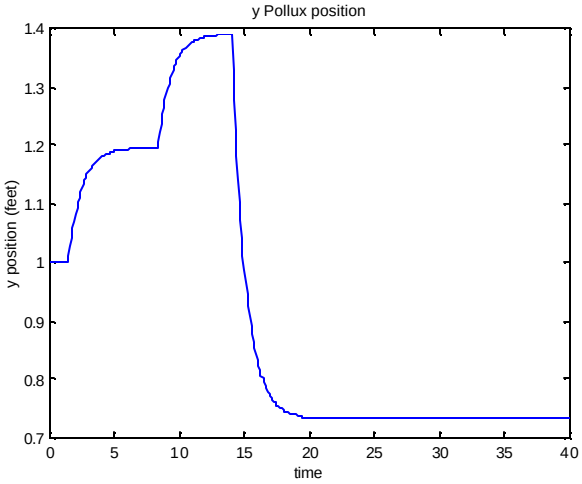


Figure 11. Y Pollux position

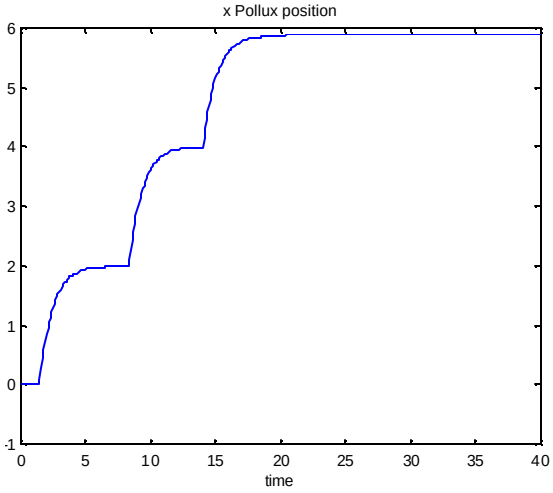


Figure 12. X Pollux position

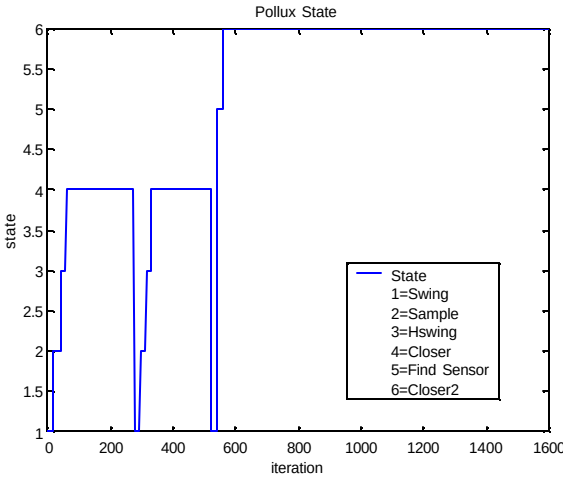


Figure 13. Pollux state

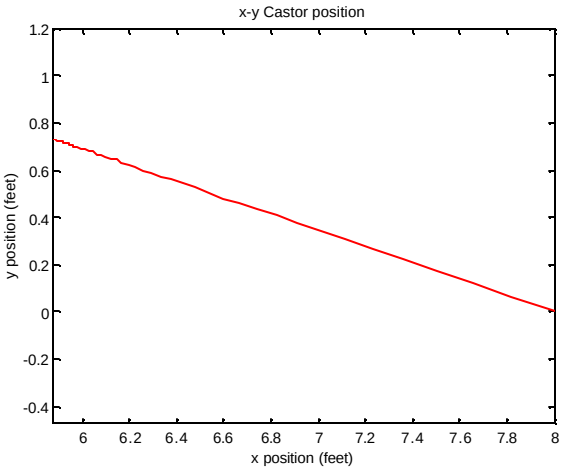


Figure 14. X-Y Castor Position

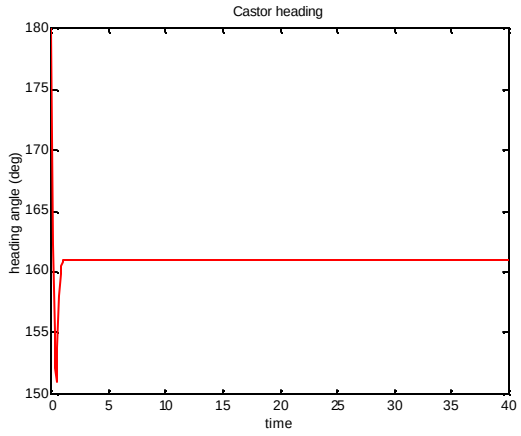


Figure 15. Castor Heading

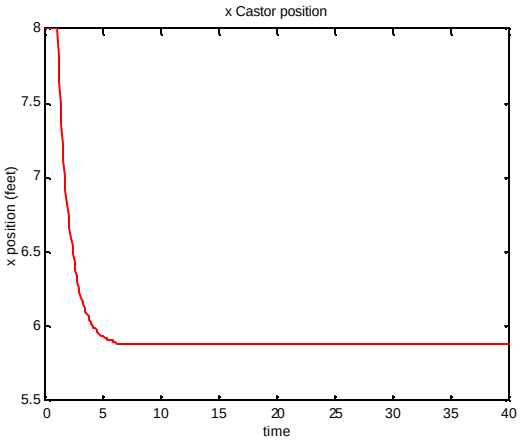


Figure 16. X Castor Position

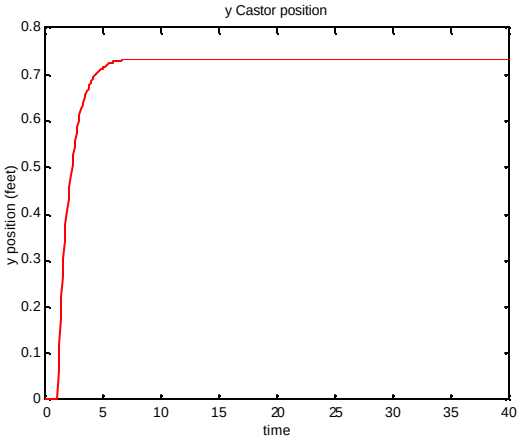


Figure 17. Y Castor Position

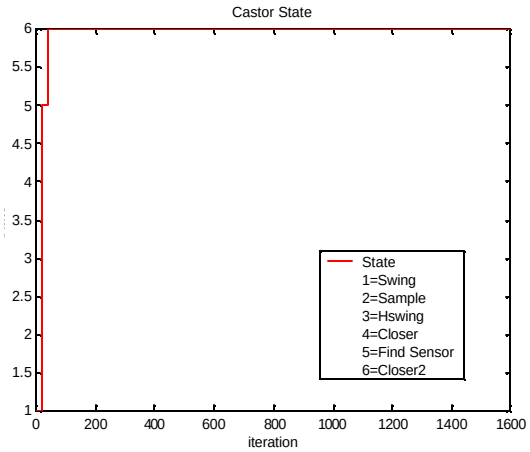


Figure 18. Castor State

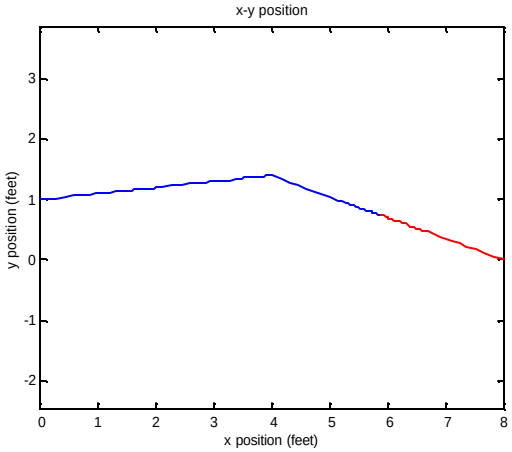


Figure 19. X-Y Position of Both Robots

Docking Limits

A more in depth use of the simulation was used to determine the actual limits of when the robots would and would not dock. Essentially a loop was placed around the already existing code to move the initial position of Pollux up and down by six-inch increments.

Figures 20, 21, and 22 are the same as in the single case results but have multiple starting positions. Although this gave a good representation of where the docking limits were, it was only for a specific angle. Because of this the simulation was run multiple times for angles zero through 40 degrees. See Figure 23 for results.

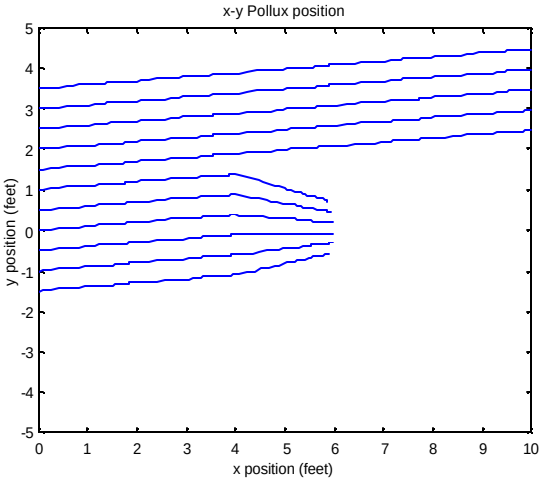


Figure 20. X-Y Pollux Position Analysis Array

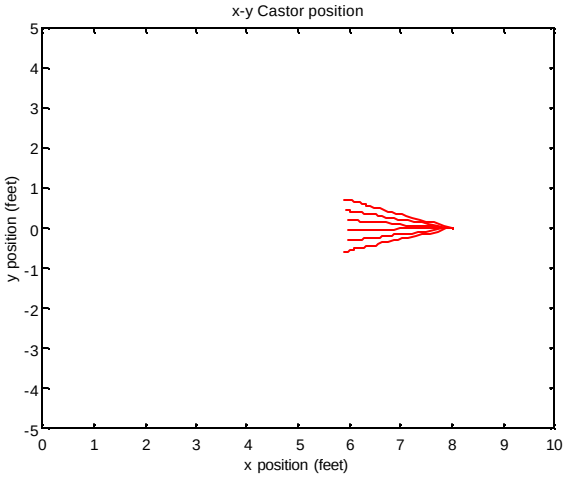


Figure 21. X-Y Castor Position Analysis Array

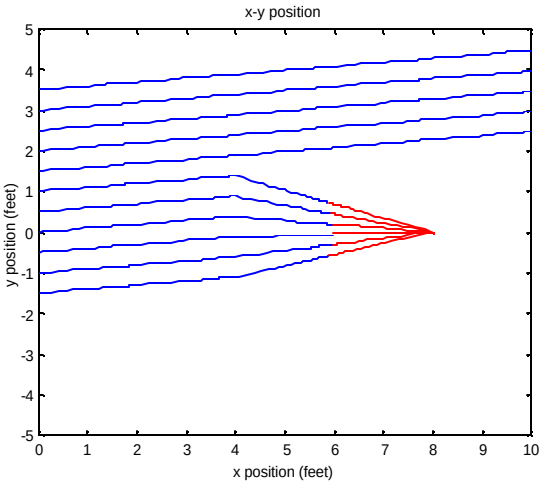


Figure 22. X-Y Position Analysis Array

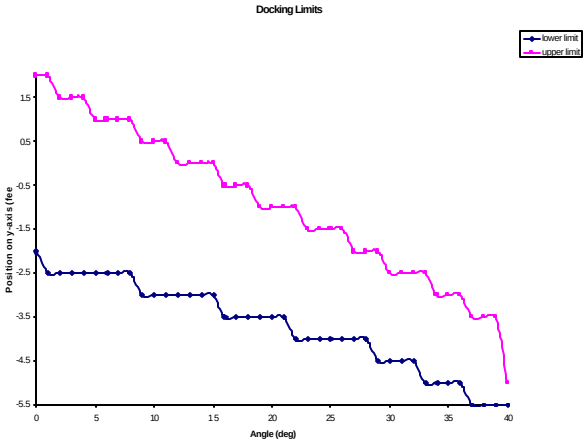


Figure 23. Docking Limits

Electronics

Using the ultrasonic transducers, one vehicle transmits at a fixed rate while the other receives. In analyzing the raw output signal from the receiving vehicle, it was noted that only the magnitude of the peak signal was the needed.

As a guide for the next section, a block diagram of the final signal conditioning circuit is shown below.

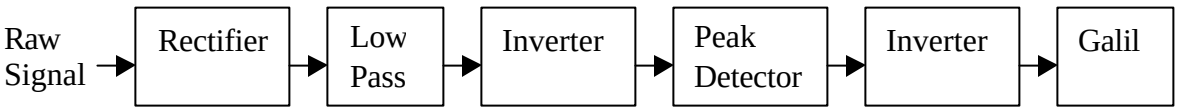


Figure 24. Signal Conditioning Circuit Block Diagram

Overall Schematic

Below is a diagram of what is going on inside the robot and how the autonomous docking is taking place. The Galil board is the brain of the operation. From there three digital outputs, two digital inputs, and three analog inputs are used.

Output 1 controls the power to the T1 acoustic development board and to the circuit which conditions the sensor output. The second actually shorts or opens a jumper on connector J1 on the T1 development board. This controls whether the ultrasonic transducer is transmiting or receiving. Output 3 controls the power to the momentary and limit switches

Analog input 1 is used for the conditioned signal output from the circuit designed for this project. Analog input 2 is used for the vehicle ID. This will be tied to a digital output on the robot's computer to determine wheather it is Castor or Pollux. This will only be needed when more than two robots are docking. For now Castor is tied to a 5V input while Pollux is tied to 0V. Analog input 3 is used for the momentary switch. The momentary switch is only pressed when both robots are fully docked. All of the analog inputs on the Galil board have an input range of ±10V. This particular board has a range of –10 V to +5.7.

The forward and reverse limit switches are plugged into a special Galil feature where it stops the motor motion when a limit is reached. In this case both switches are plugged into the forward and reverse c(z) inputs. This is because a third motor inside the robot is connected to turn the cam or mechanical lock to complete the docking sequence.

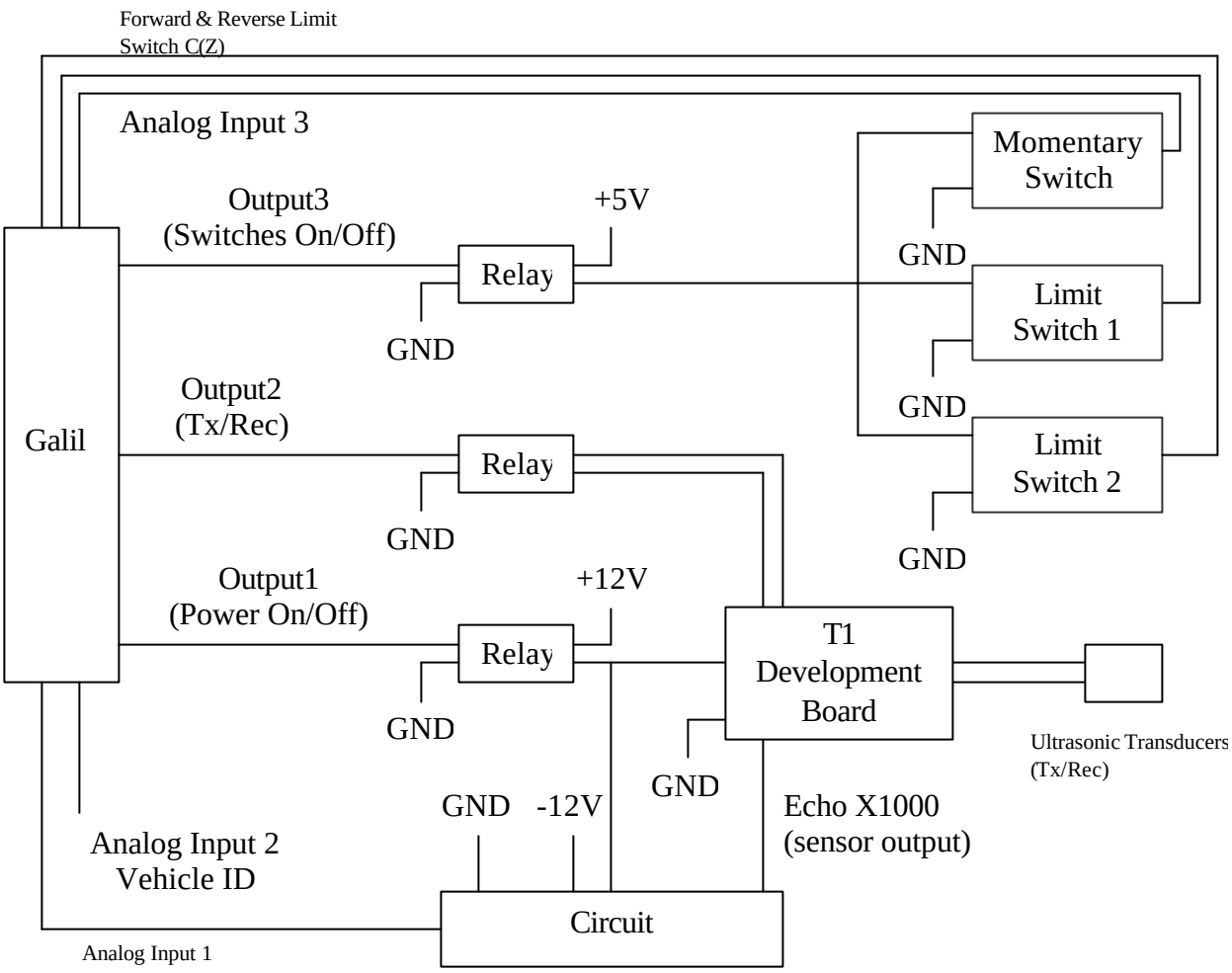


Figure 25. Overall Schematic

Physical Setup

The physical layout uses the custom designed circuit board as a type of motherboard for the system. The board contains eight Molex connectors so that each component plugs directly into it avoiding confusion. For example there is a seven-conductor ribbon cable, which is used to connect the Galil board to the circuit.

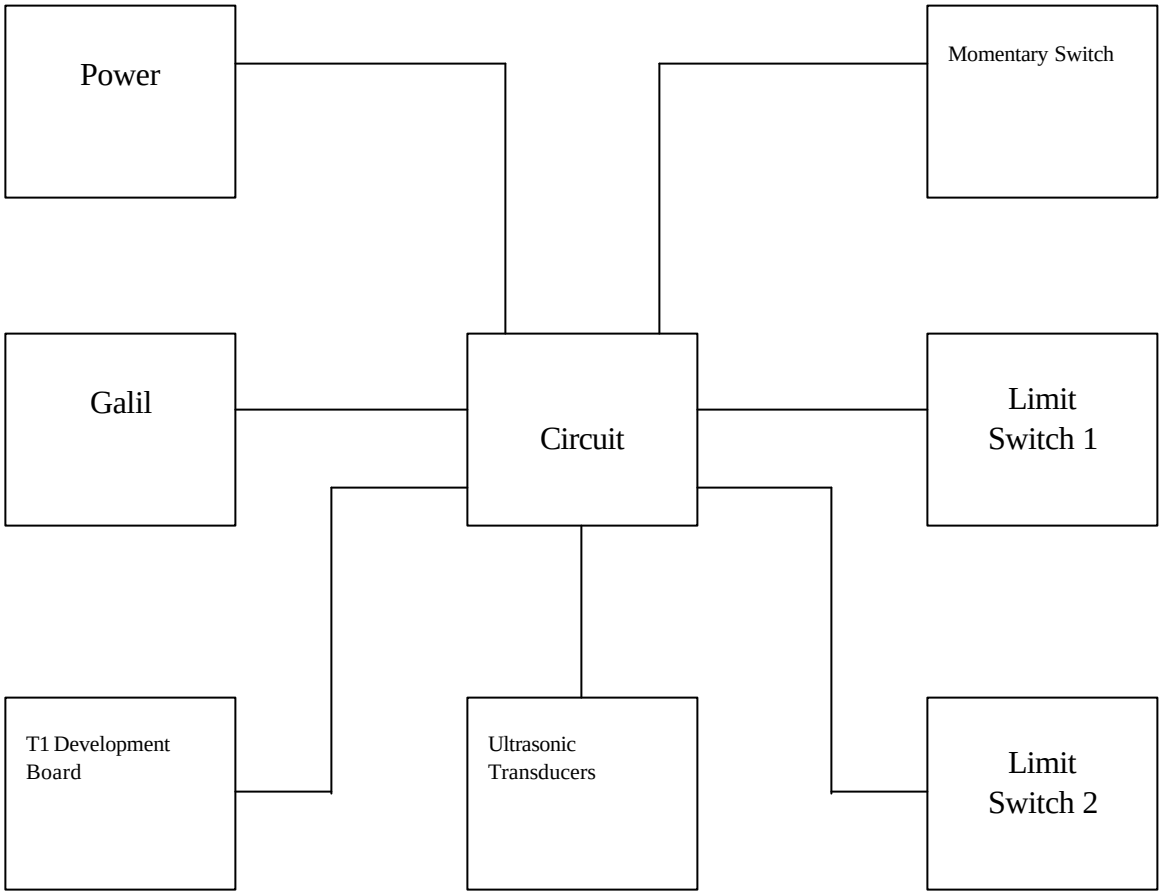


Figure 26. Physical Setup

After the circuit simulations were complete a signal-conditioning prototype was built on a breadboard. After testing and small modifications a prototype was soldered up on a protoboard. This was done a couple of times then the final circuit was sent out to be integrated. Some photographs of the different versions can be found below.

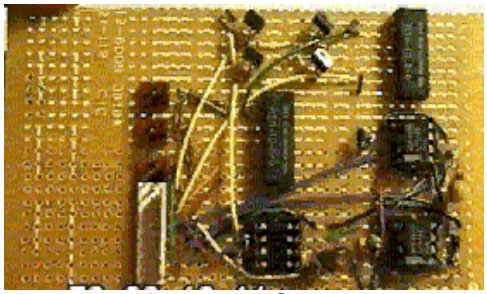


Figure 27. Circuit Hardware

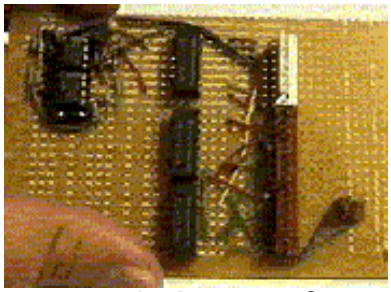


Figure 28. Circuit Hardware

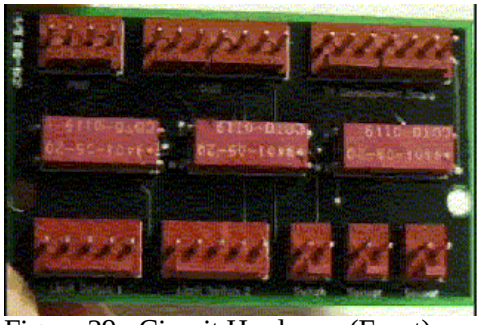


Figure 29. Circuit Hardware (Front)

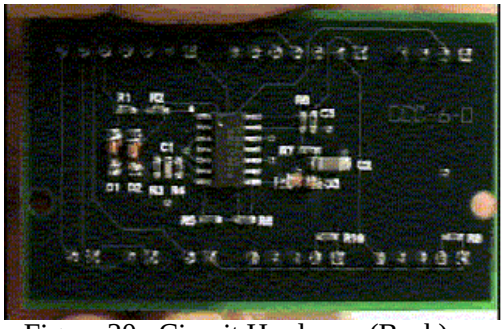


Figure 30. Circuit Hardware (Back)

Experimental Evaluation

After everything was in place testing began. The robots were autonomously docked under different circumstances. Testing was done inside and outside using versions one, two and three circuits. Some tests were run with the vehicle lids off and some tests with the lid on. Distances and angles between initial start positions were also varied.

Of all the different scenarios mentioned above only the change in angle seemed to affect the success or failure of the docking procedure. When the initial angle difference between the vehicles was too great the vehicles would not dock. This is due to the fact that the ultrasonic transducer transmits a narrow cone. This was needed to successfully align the vehicles but detracts from the docking success when there is a large compass error.

There is an obvious tradeoff. A transducer with a narrow cone aligns the vehicles more accurately while it depends more on the accuracy of both the compass and GPS units. On the other hand a wider cone output depends less on compass and GPS accuracy, but has more error in aligning the vehicles.

There is another less obvious tradeoff. The vehicles could be programmed to sample a wider angle and they would depend less on the accuracy of the compass and GPS but this would require more time and vehicle energy.

When running the autonomous docking code with the vehicles within compass and GPS specifications the vehicles docked. This suggests that the optimum docking parameters may not be set but there is enough balance in tradeoffs for a successful dock.

In conclusion both Pollux and Castor performed a successful docking routine. They were successful inside and outside. Some testing was done with external sound disturbances such as a radio playing and an airplane flying at low altitude. This did not seem to affect the docking routine in any way.

Below are some close up pictures of the nose and cone to show how both robots are mechanically linked to complete the docking routine. When the nose is inserted into the cone a momentary switch is pressed and a cam mechanically locks the two together.

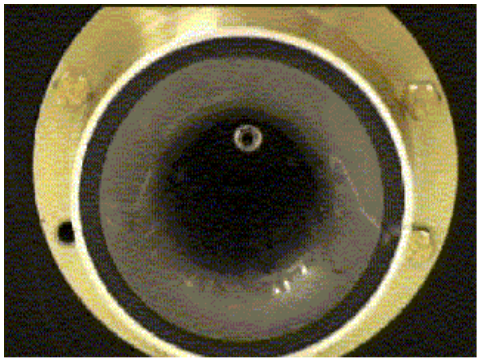


Figure 31. Castor Cone

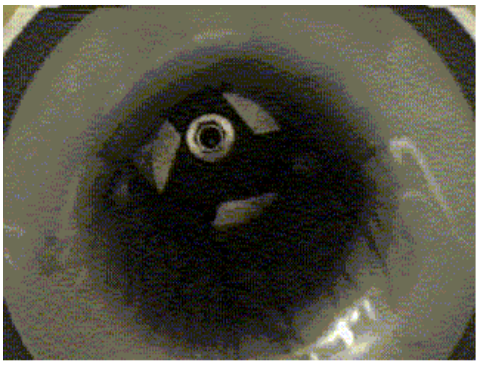


Figure 32. Castor Locking Mechanism

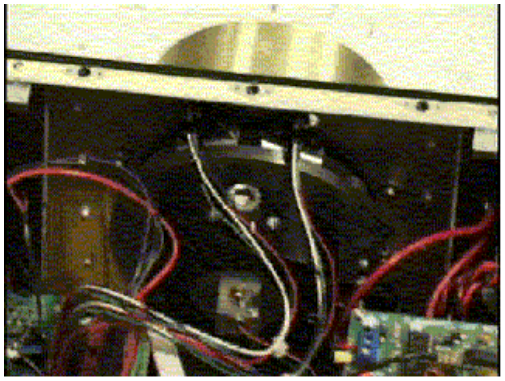


Figure 33. Castor Cone

Below are some pictures taken during a successful dock.

Figure 34. Auto-Dock in Progress

Figure 35. Auto-Dock in Progress

Figure 36. Auto-dock in Progress

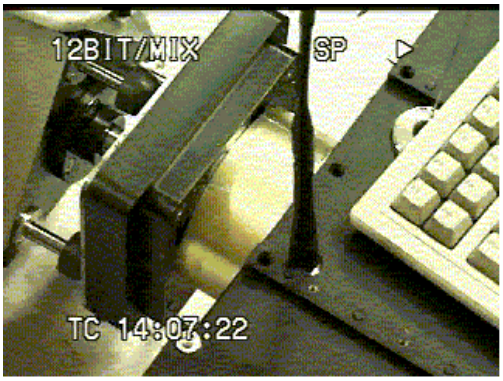


Figure 37. Auto-dock in Progress

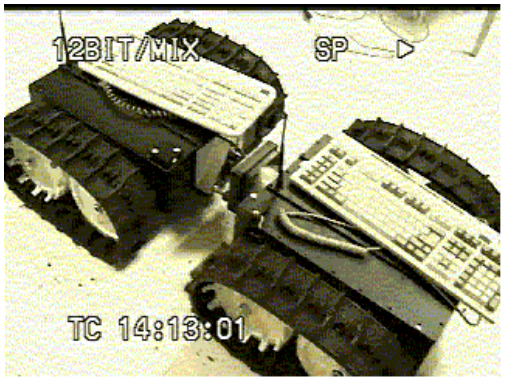


Figure 38. Castor and Pollux Docked

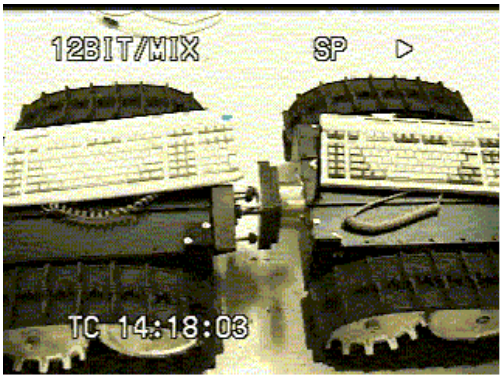


Figure 39. Castor and Pollux Docked

It can also be seen through the use of the simulation that there are limits on the initial position and heading for a successful dock to take place. In practice this will depend on the accuracy of the compass and differentially corrected GPS. If these two pieces of electronic equipment can get both robots within the determined limits a successful dock should occur. As of now, both of these pieces of equipment have not been incorporated into the robot, but specifications on both show promise of successful dock.

IV. Power Module

An additional power module was built and demonstrated. The main purpose for the power module is to extend the range of the vehicles.

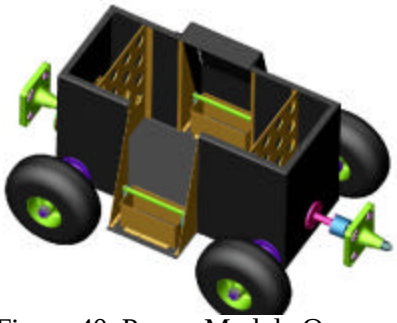


Figure 40. Power Module Open

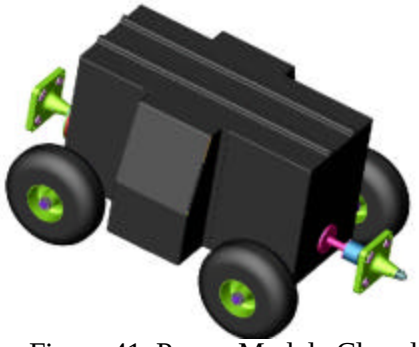


Figure 41. Power Module Closed

Take a look at the interface on the female quick disconnect (figure 42). This provides for 16 circuits between vehicles or modules. These include power transmission, actuation and general control, data transmission, and miscellaneous circuits that are unique to each desired module. The number of circuits that we have designed into the current interface is adequate for the power module and winch module. So in practice, the power module can generate enough power to run a vehicle it is connected to and charge its batteries.

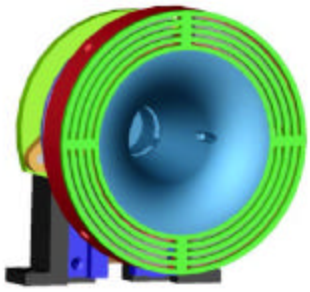


Figure 42. Female Disconnect

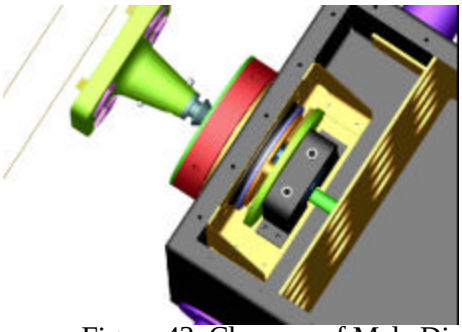


Figure 43. Close-up of Male Disconnect

It can also provide power to run additional modules. The mechanics of the interface are in keeping with the two-stage hook-up that has already been proven to be effective.

V. Software

The vehicles are controlled with the same embedded vehicle software used on the Sandia Swarm RATLER™ vehicles. This software was extended to include the release mechanism and the three linear actuators. The RATLER base-station software was also extended to include latching and release buttons and the addition controlled degrees-of-freedom. In a previous project for DARPA, we had investigated the modeling and decentralized control of multiple vehicles that do not physically interact with each other, but do interact with each other through communication. With reconfigurable robots, the vehicles physically contact each other and this interaction was added to the model of the system.

VI. 3D Study

A collaborative agreement with Dr. Daniela Rus of Dartmouth University allowed Dartmouth to perform much the 3D research. This collaboration provided a unique opportunity to exercise the planning strategy previously developed on a relatively wide three dimensional reconfigurable hardware.

A feasibility study was conducted to help guide the initial design for a self-reconfigurable system comprised of cube-shaped modules. For this study, a constraint-based self-reconfiguration planning infrastructure was prototyped that couples the geometry, kinematics, and dynamics of the robotic system and generates the space of all possible (geometrically feasible and dynamically stable) reconfigurations. In general, the planning algorithms allow one to automatically generate the task reconfiguration space by automatically determining all possible kinematic and dynamically stable next-step

reconfigurations, given an existing configuration. The strategy behind the approach is to allow randomized motion of robotic modules to take place within the influence of a potential energy field, which encourages them to move toward a minimal energy position (target configuration). The movement of cubes is determined by (1) selecting a random cube, (2) selecting a random direction for the cube to move, (3) determining whether a move is possible based on geometric constraints and system dynamics, and (4) determining whether a move will lower the total energy of the system.

Work in the 3D system combined 3D hardware systems with the configuration control algorithms to provide mission based configuration control. In collaboration with Dr Rus we have built a self-reconfiguring robot system of 4 identical modules. Each module connects to its neighbors using rotating actuators. An individual component can move in three dimensions on its neighbors. This system is currently being modified to allow global locomotion and manipulation experiments. We have built a self-reconfiguring robot system of 4 Molecules. Each module is actuated by rotations. Using this system, an individual molecule can move in plane and out of plane. We have built a self-reconfiguring robot system consisting of 9 module Crystalline Robot. Each module in this robot is actuated by expansion/contraction. The system is fully distributed, has local communication (to neighbors) capabilities and it has global sensing capabilities. We have implemented a communication infrastructure for the Crystal robot, including detecting and communicating to neighbors, sending a message around the perimeter of the robot and broadcasting a message to all the modules in the robot. We have implemented a sensing application on the Crystal robot. The modules on the perimeter of the robot sense an approaching obstacle and in response to this event they reconfigure. In a related demonstration, we showed how expansion and contraction could be used to maintain a given distance from a moving obstacle. We have developed an algorithm for distributed goal recognition in a self-reconfiguring robot. Using the Crystal hardware and the communication infrastructure developed as part of this project, we have implemented and tested this algorithm on the hardware.

In 3-D system design, the homogeneity of the system is a very stringent and difficult requirement for a designer. Each unit must have spatial symmetry. In other words, the shape and function of the robot unit must be the same for each axis of symmetry axis. To achieve a 3-D system, one needs at least three symmetry axes, and the unit must posses some degree of mobility around each axis. As a starting point of design, we selected the cube, which is one of the simplest structures with spatial symmetry. Each of three symmetry axes of the cube is orthogonal to the others, which is a great advantage in the design of mechanisms. The robotic module consists of 3 male faces that connect to 3 female faces using a ring of rare earth magnets (similar to the 2-D approach). The structure is reconfigurable by means of individual cubes traversing along the faces (along the symmetry axes) of neighboring cubes. There are two basic methods of reconfiguration: one is movement on a plane; the other is orthogonal to the plane. In both methods, the unit always makes motion accompanied with another unit.

Self-reconfiguration defines a rich class of questions not only about the robot-module design and interconnect technology (electrically and mechanically), but also about techniques required to coordinate and control motions of the subsystems. In the domain

of self-reconfiguration itself, there is a need for two kinds of planning algorithms: (1) those required to achieve a desired geometric shape and (2) those required to move globally to that resulting shape. More succinctly stated, the self-reconfiguration planning problem itself is the determination of a sequence of controllable module motions that changes the initial configuration of an overall system into a goal configuration.

Task-Reconfiguration Spaces

We began our development by attempting to characterize the tasks that could be carried out by the system (comprised by the cube-shaped robots). This involved the determination of all possible configurations the robot modules must encounter "to move" from an initial configuration to a goal configuration, as well as the determination of how to move from one configuration to the next. This characterization forms the basis of what we call the task reconfiguration space. The task reconfiguration space is, of course, dependent on the module design. While there is no universal or optimal design that accommodates all tasks, the characterization of tasks (e.g., locomotion, manipulation, static structures, etc.) lends itself towards a nominal design. For our study, the task of climbing steps was selected because of its simplicity and versatility: it is one of the simplest task that exhibits 3-D motion (motion in the plane and orthogonal to the plane) and with very few cubes. In Figure 40, individual pictures depict geometric and kinematic elements of a task reconfiguration space for climbing steps using 5 cubes with outer dimensions equal to the height of the steps.

Self-Reconfiguration Planning Approach

The approach taken in the self-reconfiguration planning allows randomized motion of the cubes to take place within the influence of a potential energy field, which encourages them to move toward a minimal energy position (target configuration). It is acknowledged that this approach partly invents a new problem; that of appropriately specifying the potential energy field for the problem. Such problems as traversal, sink to floor, and build tower require near trivial potential energy functions. Other tasks such as "build chair" or "climb stairs" require significantly more complex potential energy functions. The energy function need not be analytic and can be evaluated from a look-up table or from a geometric representation. The movement of cubes is determined by (1) selecting a random cube, (2) selecting a random direction for the cube to move, (3) determining whether a move is possible based on the constraints, and (4) determining whether a move will lower the total energy of the system.

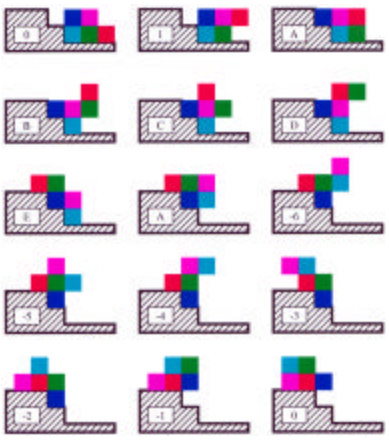


Figure 44. Task Reconfiguration Space of Cube-Shaped Robots Climbing Steps

Constraints imposed within the system are that (1) the cubes cannot occupy the same space as each other, (2) the cubes cannot occupy the same space as an obstacle, (3) the

cubes must always remain linked to the face of at least one other cube, and (4) the collection of cubes must be stable (will not topple over), before, during, and after moves. A further influence is the imposition of a function by which the total energy of the system is calculated.

Analysis capabilities that were integrated into the planning infrastructure include (1) ensuring that before and after each move the collection of cubes will be statically stable as a rigid body, (2) determining that at the beginning and end of each change in configuration the maximum acceleration rate that can be applied to the moving cubes without causing the collection of cubes to topple about one of its supporting edges, and (3) generating a finite element geometry input deck in the Exodus II format after each move. This format is used by most all of Sandia's internally developed structural, dynamic, and thermal analysis codes.

Static Stability

Before allowing a move from one reconfiguration element to a proposed next-step reconfiguration element in the task reconfiguration space, the static stability of the proposed reconfiguration is checked. The modules are not allowed to assume a configuration that is statically unstable. Static stability is considered to exist if the location of a configuration's center of mass in the x-y coordinate frame is located within the two dimensional convex hull of the points supporting the structure. This static stability test was implemented using an algorithm that determines static stability in linear time proportion to the number of support points.

Dynamic Stability (Acceleration Limits)

Requiring static stability ensures that the task reconfiguration elements (of cubes) will not topple over simply due to the acceleration of gravity. The reconfiguration elements might also topple over due to forces generated when a module or modules accelerates relative to the rest of the structure. The accelerating modules will produce reaction forces that will act on the reconfiguration element and may cause it to topple. During each configuration change a calculation is made to determine the maximum allowed acceleration rates for the moving modules both at the beginning and ending of the reconfiguration. These acceleration limits are based on incipient tipping of the reconfiguration element due to reaction forces.

Stress Analysis: The ability to write out a representation of a reconfiguration element as a finite-element geometry input deck in the Exodus II format allows the application of Sandia's powerful analysis capabilities to the reconfiguration problem. JAS3D a non-linear quasi-static solid mechanics code is the most applicable to the stress analysis of configurations (see Figure 41).

Linear Static: Using JAS3D a linear static analysis can be performed to determine the stresses between the modules in any given configuration. Gravity forces acting on the cubes and the reaction forces supporting the cubes cause these stresses. The stress distribution

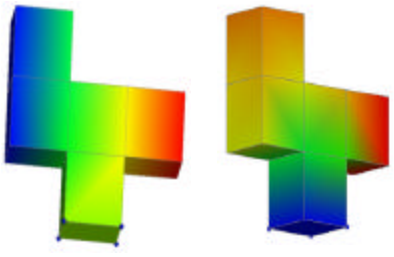


Figure 45. JAS3D Stress Calculations for a Task (Climbing Steps) Reconfiguration Element: (Left) Displacement in the Z Direction and (Right) Stress in the Z Direction

results produced by these calculations are essential for refining the design of the connections between structural modules.

Linear Quasi-Static: Using the linear quasi-static capability of JAS3D allows simulation of the stresses produced within the task reconfiguration element's structure due to acceleration of modules and the resulting reaction force. When modules move relative to each other they accelerate by exerting a force on the rest of the structure. If this force is applied to a quasi-static analysis as a dynamic load, then the resulting stresses and strains in the structure can be calculated. Because the dynamic stresses present may be significantly larger than the static stresses, the ability to calculate these stresses will be essential for refinement of the modules' design.

VII. References

[1] Gerald Recktenwald, "Numerical Methods with Matlab Implementation and Application", Prentice-Hall, pp.697-698, 2000.

[2] Benhabib, B., Zak, G., Lipton, M. G., "A generalized kinematic modeling method for modular robots," *Journal of Robotic Systems*, Vol. 6, No. 5, 1989, pp. 545-571.

[3] Cohen, R., Lipton, M. G. Dai, M. Q., Benhabib, B., "Conceptual design of a modular robot," *J. of Mech. Design*, 3,1992, pp. 117-125.

[4] Sciaky, M., "Modular robots implementation," *Handbook of Ind, Robotics*, S. Nof. ed., John Wiley & Sons, NY, 1985, pp. 759-774.

[5] Wurst, K. H., "The conception and construction of a modular robot system," *16th Int'l Symp. on Industrial Robotics*, 1986, pp. 37-44.

[6] Beni, G., "Concept of cellular robotic systems," *IEEE Int'l Symposium on Intelligent Control*, 1988, August 24-26, Arlington, VA, 1988.

[7] Beni, G., Wang, J., "Theoretical problems for the realization of distributed robotic systems," *IEEE ICRA*, 1991, pp. 1914-1920.

[8] Fukuda, T., Nakagawa, S., "Dynamically reconfigurable robotics systems," *IEEE ICRA*, 1989, pp. 1581-1586.

[9] Fukuda, T., Kawauchi, Y., "Cellular robotic systems (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," *IEEE ICRA* 1990, pp. 662-667.

[10] Yim, M., "A reconfigurable modular robot with many modes of locomotion," *JSME Int'l Conf. on Adv. Mechatronics*, 1993, pp. 283-288.

[11] Yim, M., "New Locomotion Gaits," *IEEE ICRA*, May 1994.

[12] Murata, S., Kurokawa, H., and Kokaji, S., "Self-assembling machine," *IEEE ICRA*, 1994, pp. 441-448.

[13] Murata, S., Kurokawa, H. Kokaji, S., "Self-organizing machine," *Video Proceedings, IEEE ICRA*, Nagoya, Japan, May 1995.

[14] Mark Yim, David G. Duff and Kimon D. Roufas, "PolyBot: a modular reconfigurable robot," *IEEE ICRA*, 2000, pp. 514-520.

[15] Daniela Rus, Marsette Vona., "A physical implementation of the self-reconfiguring cyrstalline robot," *IEEE ICRA*, 2000, pp. 1726-1733.

[16] Chirikjian G.S., et al, "Evaluating efficiency of self-reconfiguration in a class of modular robots," *Journal of Robotic Systems*, June 1996.

[17] Pamecha, A., Ebert-Uphoff, I., Chirikjian, G., S., "Useful metrics for modular robot motion planning," *IEEE Transactions on Robotics and Automation*, Vol. 13, No.4, 1997.

[18] Yim, M., Lamping, J., Mao, E., Chase, J.G., "Rhombic dodecahedron shape for self-assembly robots," Xerox *PARC, SPL TechReport P9710777*, 1997.

[19] Murata S., Kurakawa, H., Yoshida, E., Tomita, K., Kokaji, S., "A 3-D self-reconfigurable structure", *IEEE ICRA*, 1998.

[20] Kotay, K., Rus, D., Vona, M., McGray, C., "The self-reconfiguring robotic molecule: design and control algorithms," *Algorithmic Foundations of Robotics*, 1998.

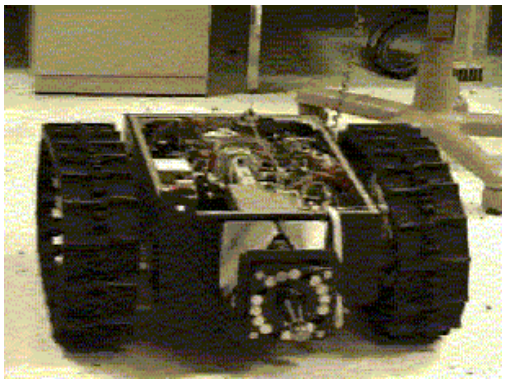
[21] Rus, D., Vona, M. "Self-reconfiguration planning with compressible unit modules", IEEE ICRA '99.

[22] Casal, A., Yim, M. "Self-reconfiguration planning for a class of modular robots," *SPIE Symp. On Intelligent Systems and Advanced Manufacturing*, Sept. 1999, Vol. 3839.

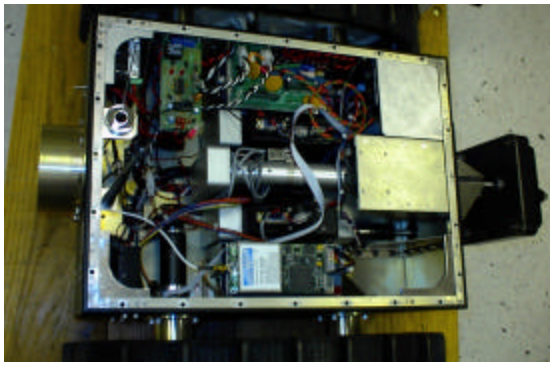
[23] K. Hosokawa, T. Tsujimori, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, and I. Endo. " Self- reoganizing collective robots with morphogene"
•http://www-robotics.cs.umass.edu/cgi-bin/robotics/
•http://www.ai.mit.edu/projects/leglab/background/links.html
•http://set.gmd.de/~worst/snake-collection_b.html
•http://avalon.epm.ornl.gov/~parkerle/sub_coop/projects.html
•http://www.epm.ornl.gov/ctrc/Research.html
•http://www-irn.sandia.gov/2nd-levels/maps-frame.html
•http://www.ornl.gov/patent/esid/1421x.htm
•http://www.chass.utoronto.ca/~reed/stage.html
•http://ece.clemson.edu/crb/research/robotics/flexible/rlfj.htm
•http://ece.clemson.edu/crb/research/flexon.htm
•http://www.sics.se/piraia/
•http://www.sics.se/~mn/dragon2.htm
•http://www.foster-miller.com/serptrus.htm
•http://ic-www.arc.nasa.gov/ic/snakebot/snake_mpg/digi/snake2.jpg
•http://ic-www.arc.nasa.gov/ic/snakebot/hello.html
•http://ic-www.arc.nasa.gov/ic/snakebot/rationale_background.html
•http://ic-www.arc.nasa.gov/ic/snakebot/snake_mpg/digi/snake1.jpg
•http://ic-www.arc.nasa.gov/ic/snakebot/snake_mpg/digi/snake2.jpg
•http://robby.caltech.edu/~chen/research.html
•http://www.ctrl.titech.ac.jp/ctrl-labs/mita-lab/snake/home.html
•http://ais.gmd.de/BAR/snake2.html
•http://www.parc.xerox.com/spl/projects/modrobots/polybot/polybot.html
•http://www.parc.xerox.com/spl/projects/modrobots/polybot/worktodo.html
•http://www.parc.xerox.com/spl/projects/modrobots/polybot/images.html
•http://www.parc.xerox.com/spl/projects/modrobots/polypod/modules.html
•http://www.parc.xerox.com/spl/projects/modrobots/polypod/polypod.html
•http://www.parc.xerox.com/spl/projects/modrobots/polypod/locomotion.html
•http://www.parc.xerox.com/spl/projects/modrobots/RD/organization.html
•http://cst-www.nrl.navy.mil/lattice/index.html
•http://www.parc.xerox.com/spl/projects/modrobots/RD/RD.html
•http://webme.ent.ohiou.edu//msysref/snake.html
•http://www.greypilgrim.com/Products.htm
•http://www.greypilgrim.com/products/platforms.htm
•http://www.greypilgrim.com/products/tools.htm
•http://www.greypilgrim.com/products/retrieval.htm
•http://www.greypilgrim.com/company/history.htm
•http://www.nist.gov/director/states/md-attach.html
•http://robotics.jpl.nasa.gov/tasks/rsi/accomplishments/snake/snake.html

VIII. Appendix

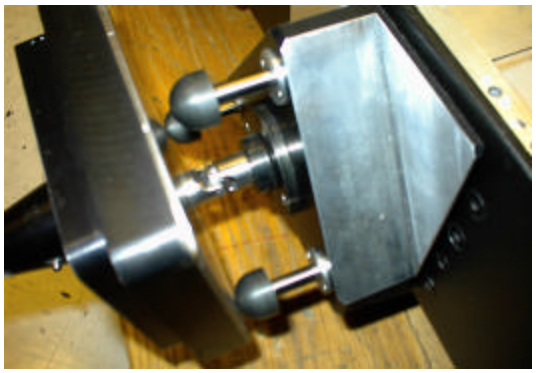
Additional Pictures of Castor and Pollux



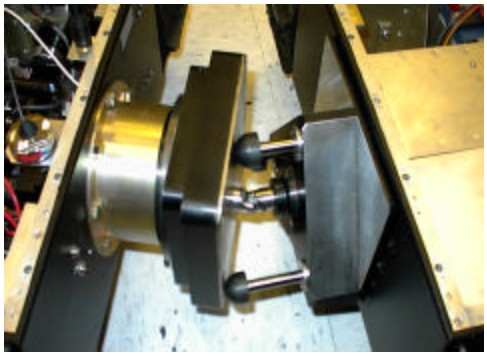
Pollux with Lid Off



Pollux



Linear Actuators

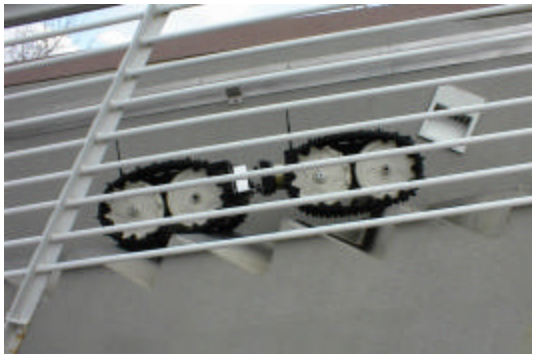


Linear Actuators

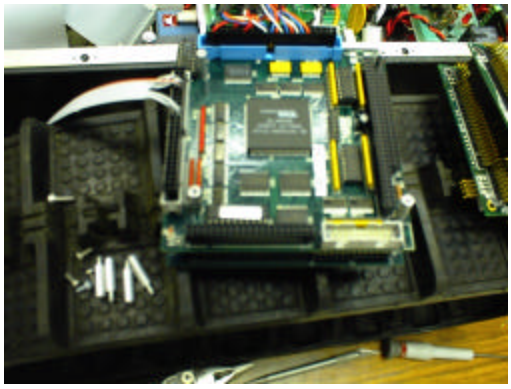
Castor and Pollux Undocked

Castor and Pollux Docked

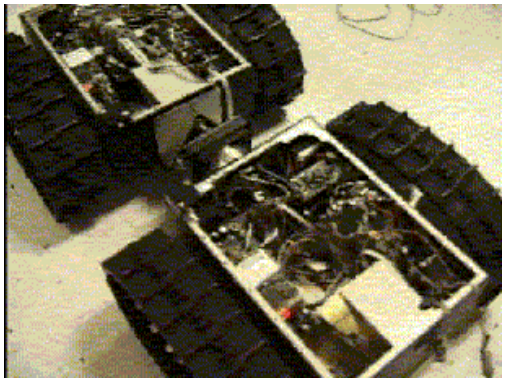
Castor and Pollux Climbing Stairs



Castor and Pollux Climbing Stairs



Galil Motion Controller



Castor and Pollux Docked

IX. Distribution

3	MS	1010	E. Hinman-Sweeney, 15222
3		1010	G. Johnston, 15222
1		1010	M. Olson, 15222
1		0879	D. Hensinger, 9231
1		1003	S. Eskridge, 15212
1		1003	K. Jensen, 15212
1		1003	J. Feddema, 15211
1		1003	D. Schmitt, 15211
1		0188	C. Meyer, 1030
5		1004	RMSEL Technical Library, 15200
1		9018	Central Technical Files, 8945-1
2		0899	Technical Library, 9616
1		0612	Review & Approval Desk, 9612 For DOE/OSTI